it leverages the power of individual user stations. The architecture of the present invention has advantages over the modem-to-modem architecture previously discussed because it supports more than two users per application. It also has advantages over the monolithic server packet network architecture previously discussed because its isochronous channels allow for both voice and real-time response and its pool of dedicated processors can be easily changed to meet changing demands unlike the monolithic server.

In addition, because the dedicated processors are heterogeneous, the developers can work in familiar environments, lower the cost of porting applications to this platform and the applications can be put onto the most appropriate platform.

FIG. **8** is a schematic of a computer system architecture according to a third preferred embodiment of the present invention. Like the architecture shown in FIGS. **6** and **7**, the system shown in FIG. **8** has a modem pool **108**, a front-end server **112** and a plurality of dedicated processors **116**. The dedicated processors **116** are divided into groups, for example, two as illustrated. Each group of dedicated processors is coupled to a LAN **130**. A switch **132** couples the modem pool **108** to the front-end server **112** and LANs **130**. In a preferred embodiment, the switch is a Router available from Cisco or Bay Networks. LANs **130** are preferably Ethernet LANs. While only two groups of dedicated processors are shown, many more may be provided.

FIG. **9** is a schematic of a computer system architecture according to a fourth preferred embodiment of the present invention. The architecture shown in FIG. **9** is similar to that shown in FIG. **8** except that the LANs have been eliminated and the dedicated processors **116** are coupled directly to a switch.

It is to be understood that depending upon the application selected one or more dedicated processors may be needed to run the application. For example, a simple game may only require one dedicated processor whereas a flight simulator application may require a plurality of dedicated processors.

It is to be understood that the forms of the present invention described herewith are to be taken as preferred examples and that various changes in the shape, size and arrangement of ports may be resorted to, without departing from the spirit of the invention or scope of the claims.

What is claimed is:

1. A method for processing real-time applications, the method comprising:

providing a front-end server;

providing a plurality of dedicated processors coupled to the front-end server so that the front-end server can communicate with at least one of the plurality of dedicated processors;

selecting at least one of the plurality of dedicated processors to execute a selected application;

transferring the selected application from a memory device to the at least one of the plurality of dedicated processors for execution;

initiating communication between a plurality of users and the at least one of the selected dedicated processors so that the plurality of users can participate in the execution of the selected application;

executing the selected application at the at least one of the selected dedicated processors; and

suspending communication between the plurality of users and the front end server.

2. A method according to claim **1** further comprising:

storing a plurality of applications in the memory device, the memory device being coupled to the front-end server; and

at the front-end server, generating appropriate communication signals to download the selected application to the at least one of the plurality of dedicated processors.

3. A method according to claim **1** further comprising:

storing applications in a memory associated with each of the plurality of dedicated processors.

4. A method according to claim **1** wherein selecting at least one of the plurality of dedicated processors includes polling the plurality of dedicated processors by the front-end server to determine which one of the plurality of dedicated processors is available to execute the selected application before that application is downloaded to the selected dedicated processor.

5. A method according to claim **1** wherein selecting at least one of the plurality of dedicated processors includes the plurality of dedicated processors communicating their status to the front-end server.

6. A method according to claim **1** wherein the plurality of dedicated processors are heterogeneous.

7. The method of claim **1** further comprising:

providing a voice bridge between one or more users of the plurality of users.

8. The method of claim **1** further comprising:

providing a voice bridge between one or more users of the plurality of users and one or more processors of the plurality of dedicated processors.

9. A method for processing real-time applications which may be executed by a plurality of users, the method comprising:

providing a front-end server that has access to a plurality of applications;

providing a plurality of dedicated processors that communicate with the front-end server, the plurality of dedicated processor being inhomogeneous;

receiving a message from at least one user of the plurality of users to the front-end server that the at least one user desires to have executed a particular application;

retrieving the particular application selected by the at least one user;

selecting a dedicated process that is of the appropriate type and capacity to run the particular application;

downloading the particular application selected by the at least one user to a memory in the selected dedicated processor;

initiating communication between the plurality of users and the selected dedicated processor; and

executing the particular application selected by the at least one user on the selected dedicated processor.

10. The method of claim **9** further comprising:

requesting at the front-end server status information from the plurality of dedicated processors; and

receiving the status information at the front-end server.

11. The method of claim **9** further comprising:

after initiating communication between the plurality of users and the selected dedicated processor, suspending communication between the plurality of users and the front-end server so that the plurality of users are communicating directly with the selected dedicated processor.

12. A computer system architecture for processing real-time applications, the architecture comprising:

a front-end server;

a plurality of dedicated processors coupled to the front-end server so that the front-end server can communicate with at least one of the plurality of dedicated processors;

a coupler communicating with the front-end server, the plurality of dedicated processors and a plurality of users, wherein one or more users communicates with the front-end server to select a selected application and the front-end server communicates with the plurality of users and at least one selected dedicated processor executes the desired application, the coupler including:

means for selecting at least one of the plurality of dedicated processors to execute the selected application; and

means for decoupling a plurality of users from the front-end server and coupling the plurality of users to the at least one of the selected dedicated processors so that the plurality of users is communicating directly with the selected dedicated processors so that the plurality of users can participate in the execution of the selected application.

13. The computer system of claim **12** further comprising a voice bridge configured to be coupled between one or more users of the plurality of users and the at least one selected dedicated processor.

14. An architecture according to claim **13** further comprising a memory coupled to the front-end server for storing a plurality of applications wherein the front-end server downloads a selected application to at least one of the plurality of dedicated processors.

15. An architecture according to claim **13** further comprising a memory for each of the plurality of dedicated processors for storing applications.

* * * * *

16. A method for processing applications, the method comprising:

providing a front-end server;

providing a plurality of dedicated processors coupled to the front-end server ;

selecting an application;

transferring the selected application from a memory device to the at least one of the plurality of dedicated processors for execution;

initiating communication between a user and the at least one of the dedicated processors so that the user can participate in the execution of the selected application;

executing the selected application at the at least one of the dedicated processors; and

suspending communication between the user and the front end server.

17. A method according to claim 16 further comprising:

storing a plurality of applications in the memory device, the memory device being coupled to the front-end server; and

at the front-end server, generating appropriate communication signals to download the selected application to the at least one of the plurality of dedicated processors.

18. A method according to claim 16 further comprising:

selecting at least one of the plurality of dedicated processors to execute the selected application.

19. A method according to claim 18 wherein selecting at least one of the plurality of dedicated processors includes polling the plurality of dedicated processors by the front-end server to determine which one of the plurality of dedicated processors is available to execute the selected application before that application is downloaded to the selected dedicated processor.

20. A method according to claim 18 wherein selecting at least one of the plurality of dedicated processors includes the plurality of dedicated processors communicating their status to the front-end server.

21. A method according to claim 16 wherein the plurality of dedicated processors are heterogeneous.

22. A method according to claim 16 further comprising:

initiating communication between at least one additional user and the at least one of the dedicated processors so that the user and the at least one additional user can participate in the execution of the selected application.

23. The method of claim 22 further comprising:

providing a voice bridge between the user and the at least one additional user.

24. The method of claim 22 further comprising:

providing a voice bridge between the user and the at least one additional user and one or more processors of the plurality of dedicated processors.

25. A method according to claim 16 wherein the selected application is a real-time application.

26. A method according to claim 22 wherein the selected application is a real-time application.

27. A method according to claim 16 wherein the selected application is a real-time game application.

28. A method according to claim 22 wherein the selected application is a real-time game application.

29. A method according to claim 25 further comprising:

executing a non-real-time application on the front-end server; and

initiating communication between the user and the front-end server so that the user can participate in the execution of the non-real-time application.

30. A method for processing real-time applications which may be executed by a plurality of users, the method comprising:

providing a front-end server that has access to a plurality of applications;

providing a plurality of dedicated processors that communicate with the front-end server, the plurality of dedicated processors being inhomogeneous;

receiving a message from at least one user of the plurality of users to the front-end server that the at least one user desires to have executed a particular application;

selecting a dedicated processor that is of the appropriate type and capacity to run the particular application;

initiating communication between the plurality of users and the selected dedicated processor; and

executing the particular application selected by the at least one user on the selected dedicated processor.

31. The method of claim 30 further comprising:

requesting at the front-end server status information from the plurality of dedicated processors; and

receiving the status information at the front-end server.

32. The method of claim 30 further comprising:

after initiating communication between the plurality of users and the selected dedicated processor, suspending communication between the plurality of users and the front-end server so that the plurality of users are communicating directly with the selected dedicated processor.

33. The method of claim 30 wherein initiating communication between the plurality of users and the selected dedicated processor comprises initiating communication between the plurality of users and the selected dedicated processor along a communication path that does not pass through the front-end server.

34. A method for processing applications which may be executed by a plurality of users, the method comprising:

providing a front-end server;

providing a plurality of dedicated processors that communicate with the front-end server and that have access to a plurality of applications, including at least one real-time application;

initiating communication between a first user and the front-end server;

sending a message from the first user to the front-end server indicating that the first user desires to have executed a particular application;

initiating communication between the first user and one of the plurality of dedicated processors through a communication pathway that does not pass through the front-end server; and

executing the particular application on the one of the plurality of dedicated processors.

35. A method according to claim 34 further comprising coupling a second user to the selected dedicated processor so that the second user may participate in the execution of the particular application.

36. A method according to claim 35 further comprising coupling additional users to the selected dedicated processor.

37. A method according to claim 34 wherein the particular application is a real-time application.

38. A method according to claim 34 wherein the particular application is a real-time game application.

39. A method according to claim 34 wherein the front-end server has access to at least one non-real-time application and further comprising executing a non-real-time application on the front-end server.

40. A method according to claim 39 further comprising:

initiating communication between the first user and the front-end server so that the first user can participate in the execution of the non-real-time application.

41. A method according to claim 34, wherein the front end server determines the status of the dedicated processors.

42. A method according to claim 41, wherein the front end server chooses an available dedicated processor to execute the particular application.

43. A computer system architecture for processing real-time applications, the architecture comprising:

a front-end server;

at least one dedicated processor coupled to the front-end server ;

a coupler communicating with the front-end server, the dedicated processor and a plurality of users, wherein one or more users communicates with the front-end server to select a selected application and the front-end server communicates with the plurality of users and at least one selected dedicated processor executes the selected application, the coupler including:

means for selecting at least one dedicated processor to execute the selected application; and

means for decoupling a plurality of users from the front-end server and coupling the plurality of users to the at least one of the selected dedicated processors so that the plurality of users is communicating directly with the selected dedicated processor so that the plurality of users can participate in the execution of the selected application.

44. The computer system of claim 43 further comprising a voice bridge configured to be coupled between one or more users of the plurality of users and the at least one selected dedicated processor.

45. An architecture according to claim 43 further comprising a memory coupled to the front-end server for storing a plurality of applications wherein the front-end server downloads a selected application to at least one dedicated processor.

46. An architecture according to claim 43 further comprising a memory coupled to the at least one dedicated processor.

47. A method for running real-time applications, the method comprising:

providing a front-end server;

providing a dedicated processor;

coupling the front end server with the dedicated processor so that the front-end server may communicate with dedicated processors;

coupling a user to the front-end server;

communicating a selection from a user device of a particular real-time application to the front-end server;

executing the particular real-time application on the dedicated processor; and

directly coupling the user device to the dedicated processor to allow the user device to participate in the execution of the particular real-time application.

48. A method for using a computer system in processing an application, the method including the steps of:

providing a front end server;

providing a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors;

connecting two users via the Internet and via the front-end server to initiate communication with the dedicated processor; and

executing a real-time application program on the dedicated processor to enable the users to communicate voice with each other.

49. The method of claim 48, wherein the step of executing the real-time application program includes facilitating a teleconference with another user.

50. The method of claim 48, wherein the step of executing the real-time application program includes forming a voice conference and connecting one of the users to the voice conference.

51. The method of claim 48, wherein the step of executing the real-time application program includes connecting one of the users to an existing voice conference.

52. The method of claim 48, wherein the step of executing the real-time application program includes forming a voice conference and enabling manipulation of a parameter of the voice conference.

53. The method of claim 48, wherein the step of executing the real-time application program includes forming a voice conference and enabling movement of one of the users from the voice conference to another voice conference.

54. The method of claim 48, further including the step of sending the user's voice stream via a telephone network.

55. The method of claim 48, further including the step of sending data with the voice.

56. A method for using a computer system in processing an application, the method including the steps of:

providing a front end server;

providing a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors; and

initiating cellular telephone communication from one of a plurality of users to the front end server to enable the dedicated processor to execute the application and facilitate communication between the one user and another of the users.

57. The method of claim 56, wherein the step of initiating includes communicating voice between the one user and another of the users.

58. The method of claim 56, wherein the step of initiating includes communicating data between the one user and another of the users.

59. The method of claim 56, wherein the step of initiating includes communicating voice and data between the one user and another of the users.

60. The method of claim 56, further including the step of: engaging in chat room discussions with the cellular telephone.

61. A method for using a computer system in communicating with an application, the method including the steps of:

providing a front end server;

providing a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors;

initiating cellular telephone communication from one of a plurality of users to the front end server to enable one of the dedicated processors to execute the application and communicate with the user.

62. The method of claim 61, further including the step of accessing the world wide web with the cellular telephone.

63. The method o f claim 61, further including the step of: communicating via the Internet with the cellular telephone.

64. The method of claim 61, wherein the step of initiating is carried out with the application program being a game application.

65. The method of claim 64, wherein the game application includes more than one user.

66. A method for using a computer system in processing an application, the method including the steps of:

providing a front end server;

providing a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors;

connecting two users via the Internet and via the front-end server to initiate communication with the dedicated processor;

executing a game application program on the dedicated processor to enable the users to play the game with each other while suspending communication between one of the users and the front end server.

67. A method for using a computer system in processing an application, the method including the steps of:

providing a front end server;

providing a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors;

connecting two users via the Internet and via the front-end server to initiate communication with the dedicated processor;

executing a game application program on more than one of the dedicated processors to enable the users to play the game with each other.

68. A method for using a computer system in processing an application, the method including the steps of:

providing a front end server;

providing a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors;

initiating cellular telephone communication from one of a plurality of users to the front end server to enable one of the dedicated processors to execute a game application program on the dedicated processor to enable the users to play the game with each other.

69. The method of any one of claims 56 through 68, wherein one of said steps is carried out with the application program being a real-time application program.

---

70. A computer system architecture for processing an application, the architecture including:

a front end server;

a plurality of dedicated processors structured so that the front end server can communicate with at least one of the plurality of dedicated processors;

a connection of two users via the Internet and via the front-end server to initiate communication with the dedicated processor; and

a real-time application program executing on the dedicated processor to enable the users to communicate voice with each other.

71. The architecture of claim 70, wherein the real-time application program facilitates a teleconference with another user.

72. The architecture of claim 70, wherein the real-time application program forms a voice conference and connects one of the users to the voice conference.

73. The architecture of claim 70, wherein the real-time application program connects one of the users to an existing voice conference.

74. The architecture of claim 70, wherein the real-time application program forms a voice conference and enables manipulation of a parameter of the voice conference.

75. The architecture of claim 70, wherein the real-time application program forms a voice conference and enables movement of one of the users from the voice conference to another voice conference.

76. The architecture of claim 70, further including a telephone network communicating the user's voice stream.

77. The architecture of claim 70, wherein the real time application sends data with the voice.

78. A computer system architecture for processing an application, the architecture including:

a front end server;

a plurality of dedicated processors structured so that the front end server can communicate with at least one of the plurality of dedicated processors; and

a cellular telephone communication from one of a plurality of users to the front end server to enable the dedicated processor to execute the application and facilitate communication between the one user and another of the users.

79. The architecture of claim 78, wherein the cellular telephone communication includes a communication of voice between the one user and another of the users.

80. The architecture of claim 78, wherein the cellular telephone communication includes a communication of data between the one user and another of the users.

81. The architecture of claim 78, wherein the cellular telephone communication includes a communication of voice and data between the one user and another of the users.

82. The architecture of claim 78, wherein the cellular telephone communication includes a chat room discussion.

83. A computer system architecture for processing an application, the architecture including:

a front end server;

a plurality of dedicated processors structured so that the front end server can communicate with at least one of the plurality of dedicated processors;

a cellular telephone communication from one of a plurality of users to the front end server to enable one of the dedicated processors to execute the application and communicate with the user.

84. The architecture of claim 83, wherein the cellular telephone communication enables accessing the world wide web.

85. The architecture of claim 83, wherein the cellular telephone communication enables communicating via the Internet.

86. The method of claim 83, the application program is a game application.

87. The method of claim 86, wherein the game application includes more than one user.

88. A computer system architecture processing an application, the architecture:

a front end server;

a plurality of dedicated processors structured so that the front end server can communicate with at least one of the plurality of dedicated processors;

a connection between two users via the Internet and via the front-end server to initiate communication with the dedicated processor;

a game application program executed on the dedicated processor to enable the users to play the game with each other while suspending communication between one of the users and the front end server.

89. A computer system architecture for processing an application, the architecture including:

a front end server;

a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors;

a connection formed between two users via the Internet and with at least one of the users via the front-end server to initiate communication with the dedicated processor;

a game application program executed on more than one of the dedicated processors to enable the users to play the game with each other.

90. A computer system architecture for processing an application, the architecture including:

a front end server;

a plurality of dedicated processors so that the front end server can communicate with at least one of the plurality of dedicated processors;

a cellular telephone communication from one of a plurality of users to the front end server to enable one of the dedicated processors to execute a game application program on the dedicated processor to enable the users to play the game with each other.

91. The architecture of any one of claims 80 through 90, wherein the application program is a real-time application program.